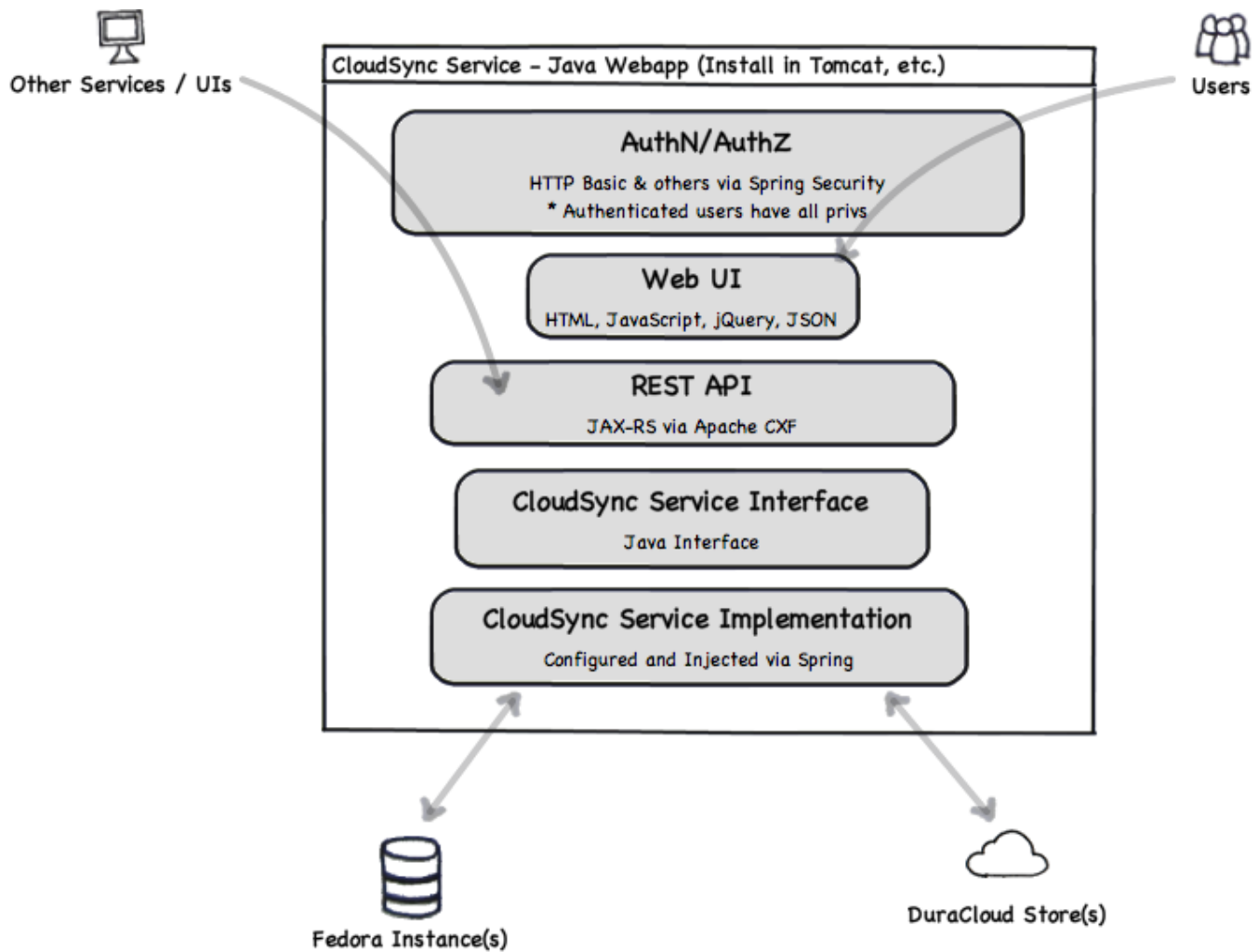


Software Design

Contents:

- [High Level Design](#)
- [Web UI Mockups](#)
- [REST API Design](#)
 - [Introduction](#)
 - [Common MIME Types, Status Codes, and Response Headers](#)
 - [Configuration](#)
 - [getConfiguration](#)
 - [updateConfiguration](#)
 - [Users](#)
 - [createUser](#)
 - [listUsers](#)
 - [getUser](#)
 - [updateUser](#)
 - [deleteUser](#)
 - [Tasks](#)
 - [createTask](#)
 - [listTasks](#)
 - [getTask](#)
 - [updateTask](#)
 - [deleteTask](#)
 - [Object Sets](#)
 - [createObjectSet](#)
 - [listObjectSets](#)
 - [getObjectSet](#)
 - [updateObjectSet](#)
 - [deleteObjectSet](#)
 - [Object Stores](#)
 - [createObjectStore](#)
 - [listObjectStores](#)
 - [getObjectStore](#)
 - [queryObjectStore](#)
 - [updateObjectStore](#)
 - [deleteObjectStore](#)
 - [System Logs](#)
 - [listSystemLogs](#)
 - [getSystemLog](#)
 - [getSystemLogContent](#)
 - [deleteSystemLog](#)
 - [Task Logs](#)
 - [listTaskLogs](#)
 - [getTaskLog](#)
 - [getTaskLogContent](#)
 - [deleteTaskLog](#)

High Level Design



Web UI Mockups

All comments (from anyone) are welcome! Please add your comments to the first mockup page listed below, as it represents the latest thinking.

- [Fedora CloudSync UI Mockup - 2011-04-19](#)
- [Fedora CloudSync UI Mockup - 2011-04-04](#)
- [Fedora CloudSync UI Mockup - 2011-03-31](#)
- [Fedora CloudSync UI Mockup - 2011-03-29](#)

REST API Design

Introduction

The CloudSync REST API is a web-accessible, resource-oriented API that exposes operations on the **key entities** of the service:

- **Configuration:** Properties that affect the global behavior of the application.
- **Users:** Users that can login to the application. Includes per-user preferences.
- **Tasks:** Define work that is to be performed or is currently being performed.
- **Object Sets:** Specifications of groups of Fedora Objects for use with tasks.
- **Object Stores:** Sources or destinations of Fedora Objects and associated Datastreams.
- **System Logs:** Raw, text/plain logs of the application. Includes everything except Task Logs.
- **Task Logs:** Raw, text/plain logs of running or completed tasks.

The exact attributes and representations of these entities have not yet been determined.

The base URL of all REST API operations is the URL of the cloudsyntax app, plus **/api/rest/**. For example:

```
http://example.org/cloudsync/api/rest/
```

Common MIME Types, Status Codes, and Response Headers

- Unless otherwise stated, requests and responses will always be in JSON or XML format. The default format is XML ("application/xml") unless ".json" is specified at the end of the URL or content negotiation is used to denote a preference for the "application/json" MIME type.
- When GET is used to retrieve information about a resource, success is indicated by a 200 OK response code.
- When POST is used to add a resource (e.g. createTask), a successful (201 Created) response will include a Content-Location header that provides the URI of the new resource. The last portion of that URI will be the newly-allocated id of that resource, which is an opaque, URI-safe string, and the response body will be the representation of the new resource, as if GET had been called on the new URI.
- When PUT is used to update a resource (e.g. updateTask), success is indicated by a 200 OK response code, and the response body will be the updated representation of the resource, as if requested via GET.
- When DELETE is used to remove a resource (e.g. deleteTask), success is indicated by a 204 No Content response code, and the response body will be empty.

Configuration

getConfiguration

```
GET /cloudsync/api/rest/configuration
```

updateConfiguration

```
PUT /cloudsync/api/rest/configuration
```

Users

createUser

```
POST /cloudsync/api/rest/users
```

listUsers

```
GET /cloudsync/api/rest/users
```

getUser

```
GET /cloudsync/api/rest/users/{id}
```

updateUser

```
PUT /cloudsync/api/rest/users/{id}
```

deleteUser

```
DELETE /cloudsync/api/rest/users/{id}
```

Tasks

createTask

```
POST /cloudsync/api/rest/tasks
```

listTasks

```
GET /cloudsync/api/rest/tasks
```

getTask

```
GET /cloudsync/api/rest/tasks/{id}
```

updateTask

```
PUT /cloudsync/api/rest/tasks/{id}
```

deleteTask

```
DELETE /cloudsync/api/rest/tasks/{id}
```

Object Sets

createObjectSet

```
POST /cloudsync/api/rest/objectsets
```

listObjectSets

```
GET /cloudsync/api/rest/objectsets
```

getObjectSet

```
GET /cloudsync/api/rest/objectsets/{id}
```

updateObjectSet

```
PUT /cloudsync/api/rest/objectsets/{id}
```

deleteObjectSet

```
DELETE /cloudsync/api/rest/objectsets/{id}
```

Object Stores

createObjectStore

```
POST /cloudsync/api/rest/objectstores
```

listObjectStores

```
GET /cloudsync/api/rest/objectstores
```

getObjectStore

```
GET /cloudsync/api/rest/objectstores/{id}
```

queryObjectStore

```
GET /cloudsync/api/rest/objectstores/{id}/objects[?set={set}&limit={limit}&offset={offset}]
```

updateObjectStore

```
PUT /cloudsync/api/rest/stores/{id}
```

deleteObjectStore

```
DELETE /cloudsync/api/rest/stores/{id}
```

System Logs

listSystemLogs

```
GET /cloudsync/api/rest/systemlogs
```

getSystemLog

```
GET /cloudsync/api/rest/systemlogs/{id}
```

getSystemLogContent

(Returns text/plain content)

```
GET /cloudsync/api/rest/systemlogs/{id}/content
```

deleteSystemLog

```
DELETE /cloudsync/api/rest/systemlogs/{id}
```

Task Logs

listTaskLogs

```
GET /cloudsync/api/rest/tasklogs
```

getTaskLog

```
GET /cloudsync/api/rest/tasklogs/{id}
```

getTaskLogContent

(Returns text/plain content)

```
GET /cloudsync/api/rest/tasklogs/{id}/content
```

deleteTaskLog

```
DELETE /cloudsync/api/rest/tasklogs/{id}
```